ACEEE Int. J. on Recent Trends in Engineering and Technology, Vol. 10, No. 1, Jan 2014

An Iterative Solution for Random Valued Impulse Noise Reduction

Manohar Koli[1] and S.Balaji[2]
[1] Dept. of Computer Science, Tumkur University, Tumkur-572103, Karnataka, India
Email: makresearch2012@gmail.com
[2] Dept. of CSE, City Engineering College, Bangalore-560061, Karnataka, India
Email: drsbalaji@gmail.com

***Abstract*— In this paper, an iterative solution for high density random valued impulse noise reduction of gray scale images is proposed. The algorithm, which works in an iterative fashion, is designed by considering the different parameters that influence the effect of noise reduction. Each iteration significantly increases the performance of the proposed algorithm. Restored Mean Absolute Error (RMAE) is used to measure and compare the performance of the algorithm. The algorithm is compared with several non-linear algorithms reported in the literature. Experimental results show that the proposed algorithm produces better results compared to the existing algorithms.**

***Index Terms*— Impulse Noise, Restoration, Image Enhancement, Image De-Noising, Adaptive Filters**

I. INTRODUCTION

Impulse noise is one which may corrupt the images during their acquisition, transmission or storage. Several algorithms have been proposed to remove impulse noise in the images. Random Valued Impulse Noise (RVIN) assumes a noise value between the minimum value 0 and the maximum value 255 of noise, as shown in equation (1).

$$X_{ij} = \begin{cases} 0 \; to \; 255 & Corrupted \; Pixel \\ X_{ij} & Non \; Corrupted \; Pixel \end{cases} \qquad (1)$$

Many value estimation techniques for corrupted pixels have been reported in literature [1]. Mean and median filters are the most popularly used techniques. In the mean filter an average value of the neighboring pixels of a scanning window is used as an estimated value of test pixel whereas in the median filter neighboring pixel values are sorted in order and then the median value is used as the test pixel value. Mean filter is effective in minimizing the mean square error value of estimation. Median filter algorithm produces good visibility in restored image. Several improved versions of mean and median algorithms have been proposed by adding new features like weight and trimming to the existing algorithms [2]. In non-linear filters conditions are used for separating or identifying noisy pixels from uncorrupted pixels and only corrupted pixels are restored. Some of the popular approaches are: (i) global and local thresholds calculation for separation of noisy pixels from uncorrupted pixels, (ii) adaptive median value calculation for accurate restoration or replacement value calculation, (iii) weight based restoration for combining or considering the effect of more than one affecting features of noise replacement, (iv) progressive or iterative replacement to improve the accuracy of algorithm by repetitive identification and replacement of noisy pixels, (v) two or multi-phase algorithms wherein distinct unrelated stages are used to avoid propagation of noise signal by calculating exactly corrupted and

DOI: 01.IJRTET.10.1.10

uncorrupted pixels and considering only the uncorrupted pixels for restoration value calculation, (vi) switching techniques to combine more than one existing filters to use different efficient filters in different noisy conditions since no single filter best fits for all noise levels. Some algorithms are good for low noise ratio. Soft computing techniques such as fuzzy, neural, edge preserving and decision based techniques are also reported for enhancement of efficiency of algorithms and also produce the good visibility.
In literature, it is observed that only few algorithms are proposed to handle RVIN. Our main aim is to provide a better solution to RVIN than the available algorithms in literature. Hence, our Proposed Random Valued Impulse Noise Algorithm (PARVIN) is compared with Adaptive Median Filters (AMF) [3], Progressive Switching Median Filter (PSMF) [4], Tri-State Median Filter (TSMF) [5], Adaptive Fuzzy Switching Filter (AFSF) [6], a New Impulse Detector Based on Order Statistics Filter (NIND) [7], An Efficient Algorithm for the Removal of Impulse Noise from Corrupted Images (AEAFRIN) [8], a New Fast and Efficient Decision-Based Algorithm(DBA) [9], An Improved Adaptive Median Filter (IAMF) [10], Robust Statistics Based Algorithm (RSBA) [11], Decision Based Adaptive Median Filter (DBAF) [12], Image Restoration in Non-linear Filtering Domain Using MDB Approach (MDBF) [13], Detail Preserving Adaptive Filter (DPAF) [14] and A Universal Denoising Framework (UDF) [15].

II. METHODOLOGY

Our algorithm is designed considering the fact that linear noise removing algorithms work well in removing the impulse noise. But, since the efficiency of the algorithm decreases and also blurs the images, the algorithm is applied linearly to all pixels including the non-noisy ones. Hence, while applying the linear filter we identify the corrupted pixels and we apply the filter only for the corrupted pixels. In our algorithm, output of the linear filter and the mean threshold are used for identifying the corrupted noisy pixels. The order in which we replace the noisy pixels also plays an important role in noise suppression. To improve the efficiency of the algorithm, corrupted pixels are replaced in the order of highly corrupted to less corrupted pixels. This way we are able to control the propagation of the noise signal. Our method has three steps (algorithms Alg-1, Alg-2 and Alg-3). Complete block diagram of the proposed system is shown in figure (1).

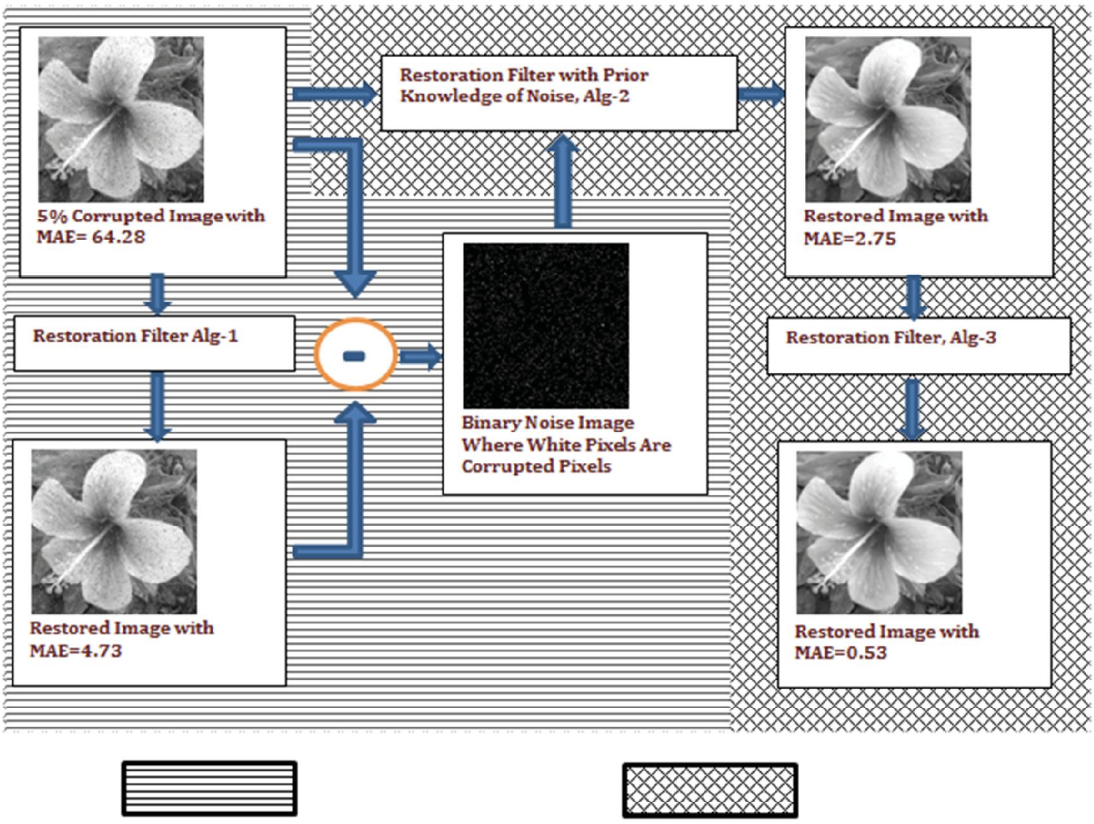
Algorithm Alg-1

1. Read corrupted image I and detected binary noise image Bn.
2. Calculate size of input image $[x, y] = size\ (I)$ and initialize new restored image $R = I$;
3. Calculate value of $s = (sum\ (Bn)/(x * y) * 10) + 1$.
4. Initialize variable $k = 0$.
5. Repeat steps 5 through 9 while $k <= s$ else go to step 10.
6. Increment $k = k + 1$;
7. Scan input image I using 3x3 window using the variables i and j where i indicates the row number and j indicates the column number such that the centre pixel of window (3, 3) overlaps the test pixel $I(i, j)$.
8. Sort all nine scanned pixels of window in ascending order a1 to a9.
9. If test pixel $I(i, j)$ is less than or equal to $a(k)$ or greater than or equal to $a(10 - k)$ then replace pixel $R(i, j)$ with $a(5)$.
10. Copy restored image R to I, $(I = R)$.

In step one corrupted pixels are identified by comparing the output of the linear restoration filter proposed in **Alg-1** and the original image. Mean threshold is used to generate binary noise image Bn where the pixel value one indicates corrupted pixel and zero indicates non-corrupted pixel. In step two, the corrupted image is restored with the help of the noisy pixels present in Bn. **Alg-2** is used for initial restoration of the noisy pixels. Once initial noise suppression is completed **Alg-3** is used for complete restoration of the corrupted image.

Algorithm Alg-2

1. Read corrupted image I.
2. Calculate size of input image $[x, y] = size\ (I)$ and initialize new restored image $R = I$;
3. Initialize binary noise image $Bn = zeros(x, y)$ and variable $k = 0$.
4. Repeat steps 4 to 8 while $k <= 4$ else go to step 9.
5. Increment $k = k + 1$;
6. Scan the input image I using 3x3 window using variables i and j where i indicates the row number and j indicates the column number such that the centre pixel of window (3, 3) overlaps the test pixel $I(i, j)$.



Noise Detection

Noise Reduction

Figure1. Block Diagram of Proposed System

7. If corresponding pixel of test pixel in Bn image is 1 then select window pixels where corresponding value of pixel in Bn is 0.
8. If the numbers of selected pixels are more than 2 then $R(i, j)$ value is replaced by median value of selected pixels else increase the window size and repeat steps 7 and 8.
9. Calculate difference of R and I, and store in D.
10. Calculate m mean of D. Convert D to binary noise image Bn using threshold m. Pixels having intensity value more than threshold value are considered as 1 else 0.

Algorithm Alg-3

1. Read corrupted images I.
2. Calculate size of the input image $[x, y] = size(I)$ and initialize new restored image $R = I$;
3. Calculate value of $s = (sum\ (Bn)/(x * y) * 10) + 1$.
4. Initialize variable $k = 0$.
5. Repeat steps 5 to 9 while $k <= s$ else go to step 10.
6. Increment $k = k + 1$;
7. Scan the input image I using 3x3 window using variables i and j where i indicates the row number and j indicates the column number such that centre pixel of window (3, 3) overlaps the test pixel $I\ (i, j)$.
8. Sort all nine scanned pixels of window in ascending order $a1$ to $a9$.
9. If test pixel $I(\ i, j)$ is less than equal to $a(k)$ or greater than equal to $a(10 - k)$ then replace pixel $R(\ i, j)$ with $a(5)$.
10. Display restored image R.

III. Performance Measurements

To evaluate performance of our proposed algorithm, four different natural images (IMAGE-1, IMAGE-2, IMAGE-3, IMAGE-4) are used. The performance is measured using RMAE. Figure 2 and Figure 3 show

restoration results of our algorithm for the images IMAGE-1 and IMAGE-2 for different amounts of noise ratio. Visibility of output of 70% noisy image clearly shows that the efficiency of our algorithm is very high. Figure 4 and Figure 5 show restoration results of different filters. The visibility of the outputs clearly shows that efficiency of our algorithm is high compared to other algorithms. Calculated RMAE for image IMAGE-3 and IMAGE- 4 are shown in Table 1 and Table 2. Compared to other popular algorithms RMAE value of our algorithm is very high. Graphical analyses of results are shown in Figure 6 and Figure 7.

$$RMAE = 1 - \frac{(MAE\ of\ Corrupted\ Image - MAE\ of\ Restored\ Image)}{MAE\ of\ Corrupted\ Image} \quad (2)$$

$$MAE \quad = \quad \frac{\sum_i^M \sum_j^N (X_{ij} - R_{ij})}{(M \times N)} \quad (3)$$

Where

- X - Original Image.
- R - Restored Image.
- M X N - Size Of Image.
- MAE - Mean Absolute Error.
- RMAE - Restored Mean Absolute Error.

IV. Simulation And Results

To evaluate performance of our proposed algorithm, four different natural images (IMAGE-1, IMAGE-2, IMAGE-3, IMAGE-4) are used. The performance is measured using RMAE. Figure 2 and Figure 3 show restoration results of our algorithm for the images IMAGE-1 and IMAGE-2 for different amounts of noise ratio. Visibility of output of 70% noisy image clearly shows that the efficiency of our algorithm is very high. Figure 4 and Figure 5 show restoration results of different filters. The visibility of the outputs clearly shows that efficiency of our algorithm is high compared to other algorithms. Calculated RMAE for image IMAGE-3 and IMAGE- 4 are shown in Table 1 and Table 2. Compared to other popular algorithms RMAE value of our algorithm is very high. Graphical analyses of results are shown in Figure 6 and Figure 7.

TABLE I. RMAE VALUES OF FILTERS FOR RVIN IMAGE-3(230X230)

NOISE RATIO ► **FILTERS ▼**	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
AMF	17.22	33.34	34.59	32.24	28.38	25.65	22.21	19.23	16.46
PSMF	-75.76	5.99	30.5	41.49	46.41	47.55	44.26	39.47	33.94
TSMF	-197.5	-52.89	-6.38	18.07	29.46	34.91	36.9	36.11	34.55
AFSF	-73.79	2.06	24.1	33.44	36.78	36.9	35.26	32.58	29.6
NIND	-13.52	31.43	45.1	51.3	52.89	49.71	43.8	35.44	25.68
AEAFRIN	-70.85	-0.39	24.09	33.99	37.7	37.02	35.26	33.06	29.61
DBA	-2.62	-0.58	-0.05	0.14	0.45	0.42	0.56	0.71	0.57
IAMF	-47.46	17.77	34.71	42.2	41.71	33.41	17.66	7.68	3.6
RSBA	17.3	34.09	34.36	32.51	29.15	25.64	22.11	18.95	15.86
DBAF	-62.19	7.17	27.66	36.13	38.6	38.02	35.95	32.99	29.38
MDBF	17.03	33.44	34.36	32.24	29.2	25.55	22.35	19.22	16.91
DPAF	16.83	32.64	33.3	32.29	29.24	25.42	22.16	19.34	16.63
UDF	-133.8	-25.76	8.7	25.76	34.77	37.55	39.03	36.76	34.31
PARVIN	80.18	78.71	75.57	73.91	70.23	66.97	61.92	55.17	44.2

TABLE II. RMAE VALUES OF FILTERS FOR RVIN IMAGE-4 (300X300)

NOISE RATIO ► FILTERS ▼	10	20	30	40	50	60	70	80	90
AMF	57.52	53.27	46.73	39.64	32.5	25.99	20.39	15.78	12.35
PSMF	48.87	67.78	73.11	74.62	72.02	64.45	51.27	37.13	26.19
TSMF	-35.2	29.84	51.83	58.9	57.19	50.78	41.11	32.01	25.53
AFSF	40.17	60.35	62.68	58.53	51.74	43.06	35.02	28.28	22.09
NIND	60.54	74.53	79.58	80.99	79.68	74.12	61.7	42.57	24.13
AEAFRIN	41.81	61.93	65.64	62.91	55.59	46.17	36.88	28.7	22.24
DBA	0.6	1.05	1	0.96	0.91	0.84	0.84	0.66	0.61
IAMF	52.97	68.36	71.2	72.13	63.14	39.24	17.68	7.15	2.61
RSBA	56.51	53.3	46.95	39.15	32.21	25.94	20.43	16.22	12.31
DBAF	49.28	66.98	69.34	64.3	57.64	47.72	37.39	28.75	22.46
MDBF	57.4	55.14	47.19	39.97	33.01	26	20.39	15.85	12.17
DPAF	57.78	54.49	47.57	40.28	33.13	25.77	20.67	15.91	12.59
UDF	18.25	54.53	66.17	70.68	70.77	66.14	55.14	42.62	33.38
PARVIN	93.6	92.9	91.2	90.19	87.46	82.27	71.65	54.13	37.12

05% NOISE RATIO

20% NOISE RATIO

ORIGINAL IMAGE -1 (280X280)

RESTORED IMAGE OF 05% NOISE RATIO RMAE= 93.02

RESTORED IMAGE OF 20% NOISE RATIO R MAE= 89.02

35% NOISE RATIO

50% NOISE RATIO

65% NOISE RATIO

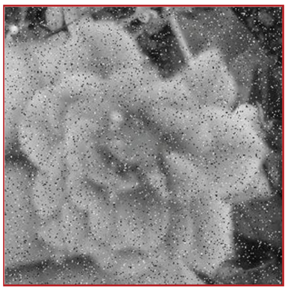
RESTORED IMAGE OF 35% NOISE RATIO RMAE= 86.76

RESTORED IMAGE OF 50% NOISE RATIO RMAE= 83.58

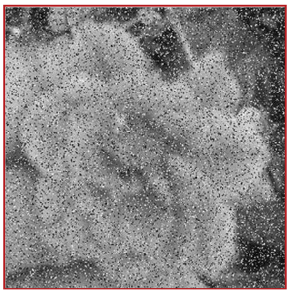
RESTORED IMAGE OF 65% NOISE RATIO RMAE= 75.24

Figure 2. Restoration Results Of Images-1upto 65% Of Noise Ratio

ORIGINAL IMAGE-2 (290X290)



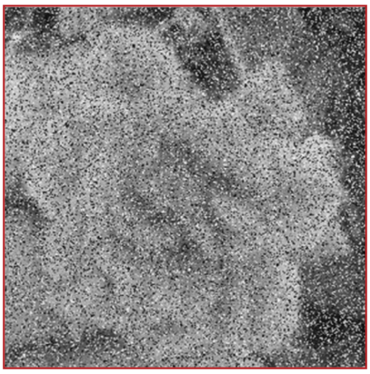
10% NOISE RATIO



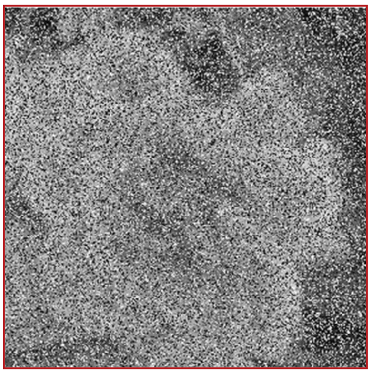
25% NOISE RATIO

RESTORED IMAGE OF10% NOISE RATIO RMAE= 99.27

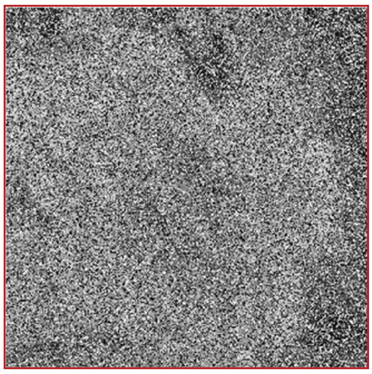
RESTORED IMAGE OF 25% NOISE RATIO RMAE= 98.58



40% NOISE RATIO

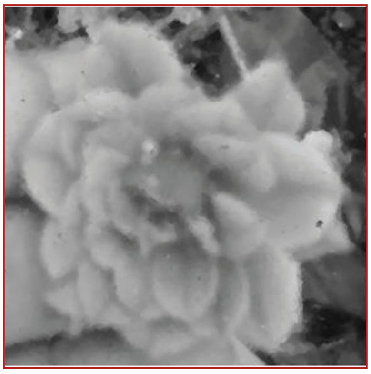


55% NOISE RATIO

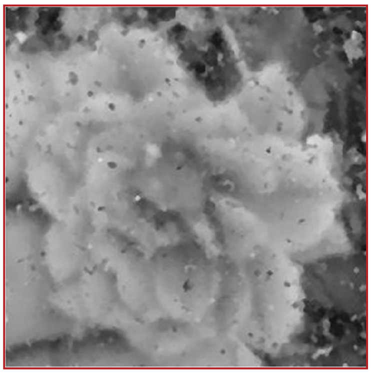


70% NOISE RATIO

RESTORED IMAGE OF40% NOISE RATIO
RMAE= 97.57



RESTORED IMAGE OF 55% NOISE RATIO
RMAE= 95.74

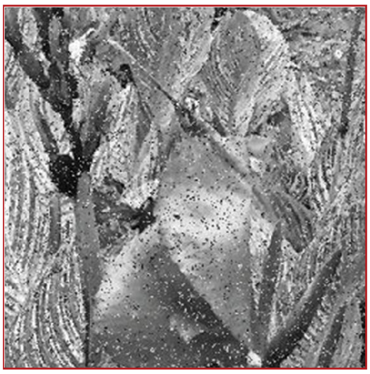


RESTORED IMAGE OF70% NOISE RATIO
RMAE= 89.59

Figure 3. Restoration Results Of Images-2 Upto 70% Of Noise Ratio

ORIGINAL IMAGE-3 (260X260)

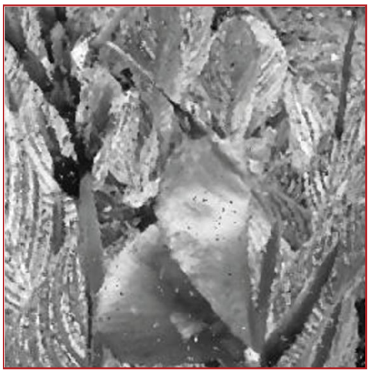
PARVIN



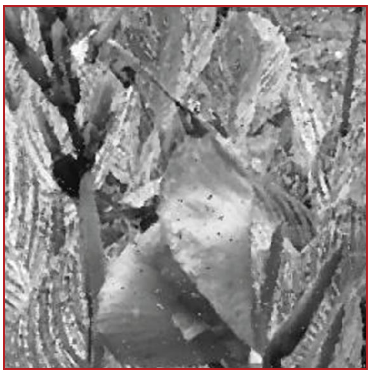
AMF

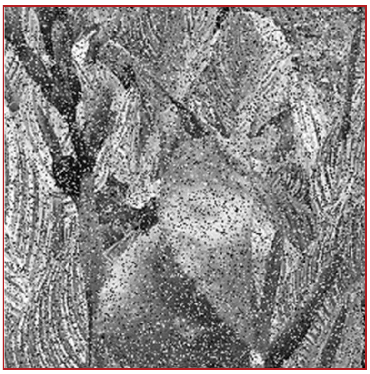
PSMF

TSMF

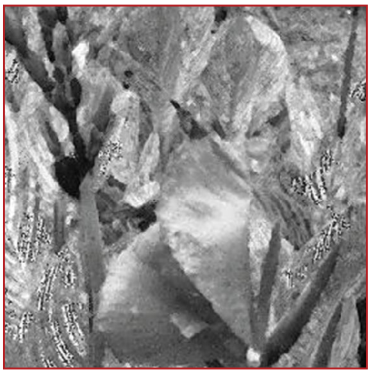
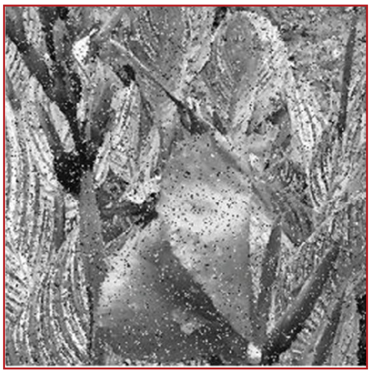
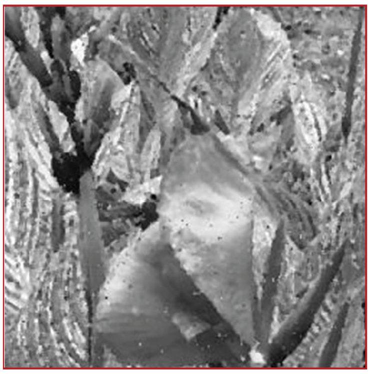


AFSF

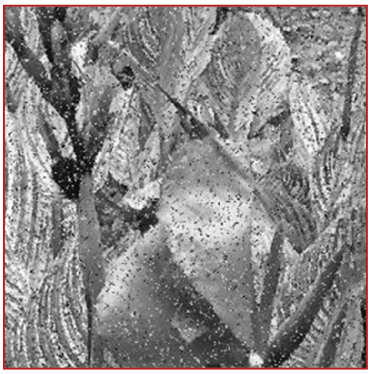
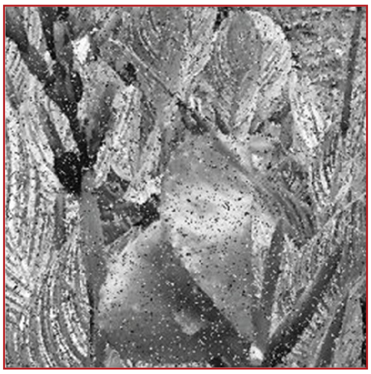




NIND **AEAFRIN** **DBA**

IAMF **RSBA** **DBAF**


MDBF **DPAF** **UDF**

Figure 4.Results Of Filters For Image-3 (260x260) With 30% Rvin



ORIGINAL IMAGE-4 (300X300) **PARVIN** **AMF**



PSMF **TSMF** **AFSF**

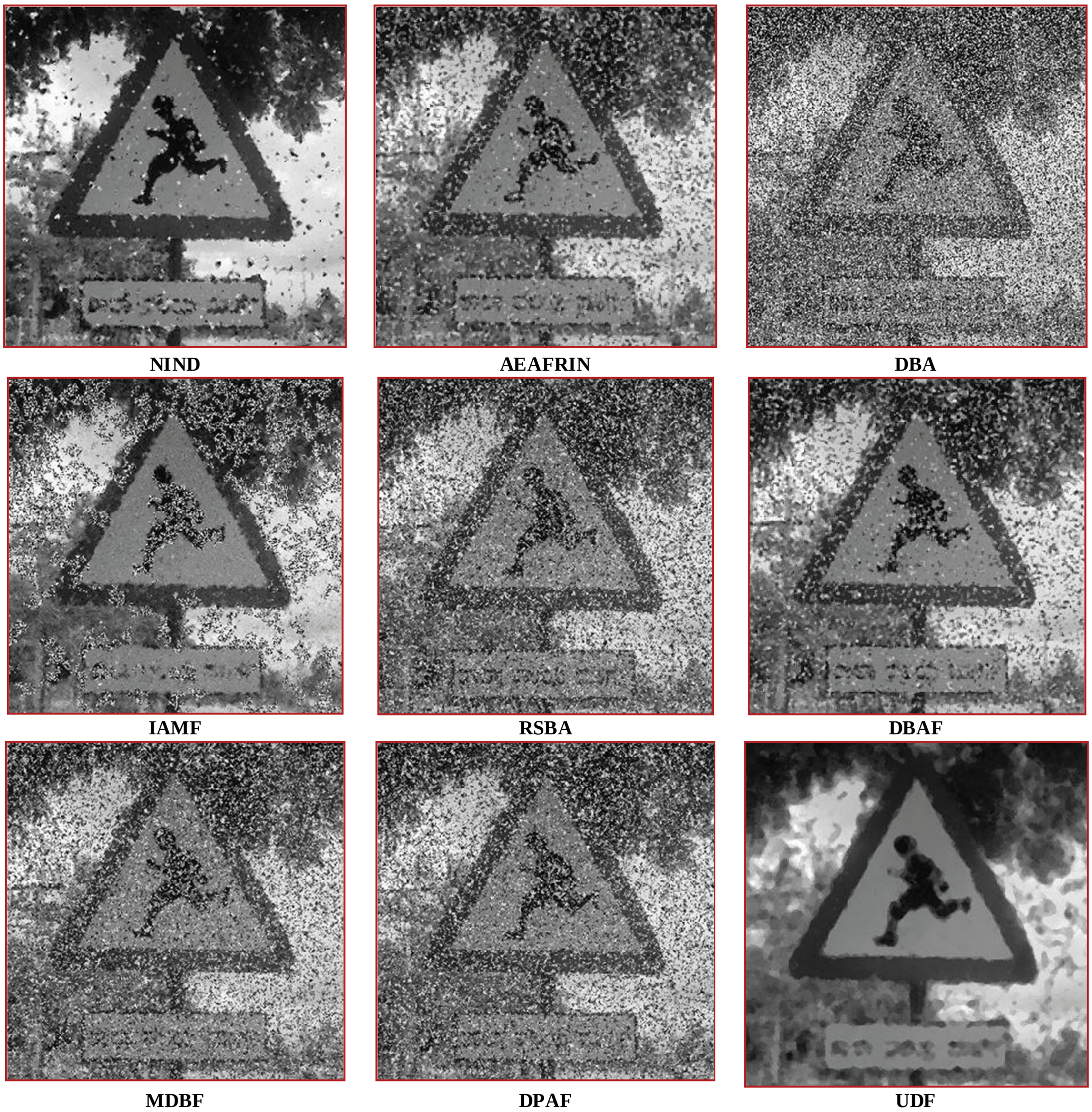


Figure 5.Results Of Filters For Image-4 (300x300) With 70% Rvin

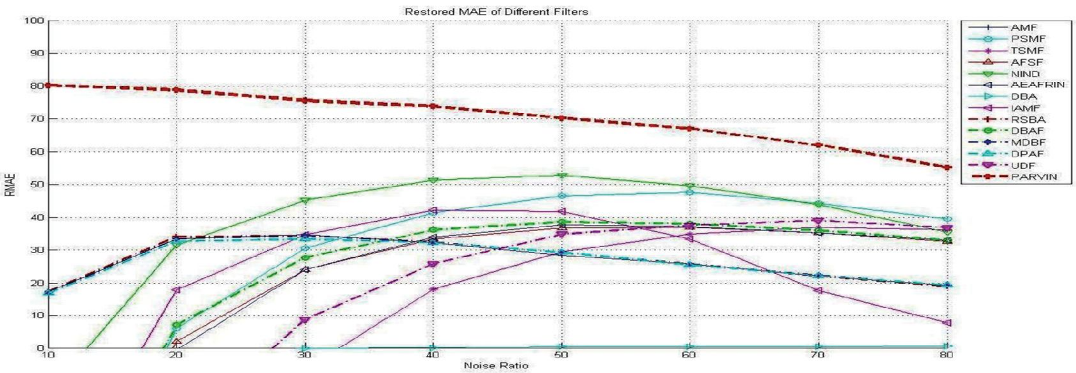


Figure 6. Rmae For The Image-3(260x260) With Rvin

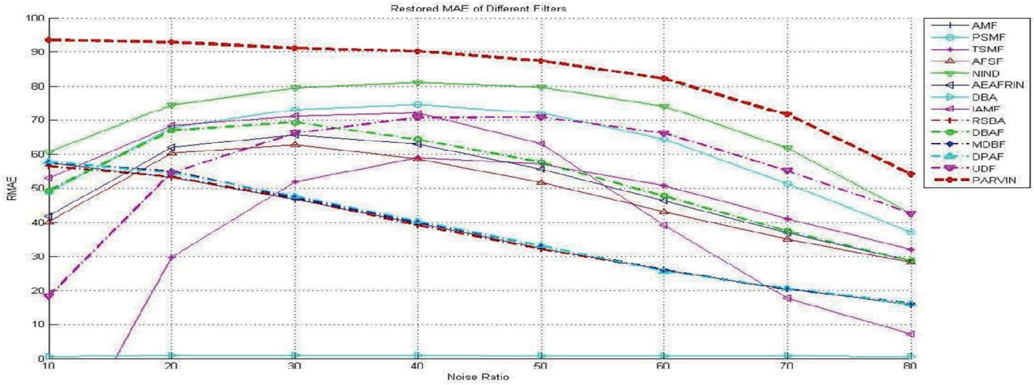


Figure7. Rmae For The Image-4 (300x300) With Rvin

V. CONCLUSIONS

In this paper, an efficient iterative method to remove random valued impulse noise from gray scale image is proposed. Each iteration of the method significantly increases the quality of input image and the proposed algorithm controls the flow of noise signal apart from producing consistent and very high quality output. Experimental results show that efficiency of the algorithm is very high compared to other popular algorithms reported in the literature. Further, the proposed algorithm works well in both the low and the high noise ratio up to 70%. This algorithm is a promising solution for RVIN reduction as it maintains consistency in performance. Study of the suitability and performance of the proposed algorithm for other types of noise and images is part of our future work.

REFERENCES

[1] Manohar Annappa Koli ,"Review of Impulse Noise Reduction Techniques", International Journal on Computer Science and Engineering (IJCSE),Vol. 4 No. 02 February 2012, pp 184-196.

[2] Sarala singh and Ravimohan, "A review on the Median Filter based Impulsive Noise Filtration Techniques for FPGA and CPLD", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 3, March 2013,pp 821-824.

[3] H. Hwang and R. A. Haddad "Adaptive Median Filters: New Algorithms and Results" IEEE Transactions on Image Processing, Vol. 4, No. 4, April 1995, pp 499-502.

[4] Zhou Wang and David Zhang "Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images" IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing, Vol. 46, No. 1, January 1999, pp 78-80.

[5] Tao Chen, Kai-Kuang Ma, Li-Hui Chen "TriSstate Median Filter for Image Denoising" IEEE Transactions on Image Processing, Vol. 8, No. 12, December 1999, pp 1834-1838.

[6] Haixiang Xu, Guangxi Zhu, Haoyu Peng, Desheng Wang "Adaptive Fuzzy Switching Filter for Images Corrupted by Impulse noise" Pattern Recognition Letters 25 (2004) pp 1657–1663.

[7] Wenbin Luo "A New Impulse Detector Based on Order Statistics" Intl. J. Electronincs Communication (aeü) 60 (2006) pp 462–466.

[8] Wenbin Luo "An Efficient Algorithm for the Removal of Impulse Noise from Corrupted Images" Intl. J. Electron. Commun. (aeü) 61 (2007) pp 551 – 555.

[9] K. S. Srinivasan, D. Ebenezer "A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises" IEEE Signal Processing Letters, Vol. 14, No. 3, March 2007, pp 189-192.

[10] Mamta Juneja, Rajni Mohana "An Improved Adaptive Median Filtering Method for Impulse Noise Detection" International Journal of Recent Trends in Engineering, Vol. 1, No. 1, May 2009, pp 274-278.

[11] V.R.Vijaykumar, P.T.Vanathi, P.Kanagasabapathy, D.Ebenezer "Robust Statistics Based Algorithm to Remove Salt and Pepper Noise in Images" International Journal of Information and Communication Engineering 5:3 2009, pp 164-173.

[12] V.R.Vijaykumar, Jothibasu "Decision Based Adaptive Median Filter to Remove Blotches, Scratches, Streaks, Stripes and Impulse Noise in Image" Proceedings of 2010 IEEE 17th International Conference on Image Processing, September 26-29, 2010, Hong Kong, pp 117-120.

[13] S. K. Satpathy, S. Panda, K. K. Nagwanshi, C. Ardil "Image Restoration in Non-linear Filtering Domain Using MDB Approach" International Journal of Information and Communication Engineering 6:1 2010, pp 45-49.

[14] Krishna Kant Singh, Akansha Mehrotra, Kirat Pal, M.J.Nigam "A n8(p) Detail Preserving Adaptive Filter for Impulse Noise Removal" 2011 International Conference on Image Information Processing (ICIIP 2011).
[15] Bo Xiong, D. Zhouping Yin "A Universal Denoising Framework with a New Impulse Detector and Non-local Means" IEEE Transactions on Image Processing, Vol. 21, No. 4, April 2012, pp 1663-1675.